IN THE SPECIFICATION

Please replace the paragraph at page 10, line 9 to page 11, line 9, with the following rewritten paragraph:

The integer formatting part 12 performs a processing shown in FIG. 4, for example. First, it is determined whether the exponent E is equal to or greater than 150 or not, that is, whether the integer part is constituted by 23 or more bits or not (S1). If the integer part is constituted by less than 23 bits, the digit calculating part 12A determines the number of digits of the integer value ($n = E-E_0$) and outputs the result (S2). Then, the mantissa M is shifted by (23-n) bits toward the least significant position to make overflow of the fractional part occur, and "1" is placed at the (n+1)th bit position viewed from the least significant bit of the resulting integer part composed of n bits (S3). Then, "0"s are supplemented at the remaining bit positions preceding the (n+1)th bit so that the whole mantissa is composed of 23 bits, and based on the sign bit S, the resulting 23 bits are converted into a 24-bit integer in the two's complement notation, thereby providing one digital value sample in the integer format (S5). Specifically, the sign bit S is used as the most significant bit as it is. As the remaining 23 bits, if the sign S is "0" (positive), the 23 bits beginning with the least significant bit of the integer part resulting from the shifting in step S3 are used, and if the sign S is "1" (negative), the 23 bits beginning with the least significant bit of the integer part resulting from the shifting are inverted except for the most significant bit and used. If the exponent E is equal to or greater than 150 in step S1, the exponent E is limited to 150 in step S4, and then, the process continues to step S5. In the above description, the mantissa M is shifted by (23-n) bits to provide a 24-bit integer value sample in the sign and magnitude binary notation. Alternatively, most significant n ($= E-E_0$) bits may be extracted from the mantissa M, "1" be added at the top of the n bits to make the number of bits (n+1), (22-n) "0"s be added at the

top of the (n+1) bits to make the number of bits 23, and then the sign bit S be added at the top of the 23 bits to make the number of bits 24.

Please replace the paragraph at page 14, lines 2 to 15, with the following rewritten paragraph:

As described above, in the exceptional case where the ~~mantissa~~ exponent E of the digital signal sample X in the floating-point format is equal to or greater than 150 ($E \geq 150$), the integer formatting part 12 transmits an exception signal y, which indicates that the exponent E is limited to 150, to the difference producing part 14, and the difference producing part 14 produces, as a difference signal ΔX, the exponent difference (E-150), which is the difference between the value 150 of the exponent and the value of the exponent E of the digital signal sample X, and the difference for the mantissa M (in which all the bits are set at "0"). The compressing part 17 performs a lossless compression coding on the difference signal ΔX and outputs the resulting code as the difference information Cb. In this case, since all the bits of the difference for the mantissa are "0", the difference for the mantissa may not be transmitted, and only the exponent difference may be coded, and the resulting code be output as the difference information Cb.

Please replace the paragraph at page 20, lines 16 to page 21, line 4, with the following rewritten paragraph:

Fig. 8 shows a functional arrangement of a decoder 200 associated with the coder 100 shown in FIG. 6, whose parts corresponding to those of the decoder 200 shown in FIG. 5 are denoted by the same reference numerals. An expansion part 21 losslessly expansion-decodes the code sequence Ca on a divisional-unit basis, thereby producing a sequence of digital signal samples Y in the integer format. In this embodiment, an auxiliary decoding part 41

decodes the auxiliary code Cc to produce adjustment information ΔE. Based on the adjustment information ΔE, a digit adjusting part 42 performs reverse digit adjustment, by $E_i+\Delta E$, on the digital signal samples Y in the integer format. If the adjustment information ΔE is positive, the bits of each digital signal sample Y is shifted toward the most significant bit position by ΔE bits. If the adjustment information ΔE is negative, the bits of each digital signal sample Y is shifted toward the least significant bit position by ΔE bits. In this process, as with the reverse digit adjusting part 34 shown in FIG. 6, the bit positions made unoccupied by the shifting are filled with "1"s or "0"s.

Please replace the paragraph at page 21, lines 5 to 13, with the following rewritten paragraph:

The digital signal samples in the integer format from the reverse digit adjusting part 42 are converted into digital signal samples X' in the floating-point format by the floating-point formatting part 22. Each digital signal sample X' and a difference signal ΔX in the floating-point format, which is derived from the difference information Cb through reverse expansion decoding by an expansion part 23, are combined together by the combining part 24. As required, a ~~coupling~~ sample sequence generating part 43 converts the consecutive digital signals into a sample sequence, thereby reproducing the sequence of digital signal samples X in the floating-point format.

Please replace the paragraph at page 22, lines 1 to 11, with the following rewritten paragraph:

Associated with the coder shown in FIG. 9, the decoder 200 shown in FIG. 10 may reproduce the digital signal sample X in the floating-point format by first converting the digital signal sample Y in the integer format, which has been losslessly expansion-decoded

by the expansion part 21, into a digital signal sample in the floating-point format by the floating-point formatting part 22, combining together the resulting digital signal sample and the difference signal ΔX in the floating-point format losslessly expansion-decoded by the expansion part 23 by the combining part 24, and then, adjusting the digit of the exponent E of the combination digital signal by the reverse digit adjusting part 42 using the adjustment information ΔE decoded by the auxiliary decoding part 41.

Please replace the paragraph at page 24, lines 24 to page 25, line 20, with the following rewritten paragraph:

Such a down-sampling part ~~41~~ 36 is provided in the case where it can improve the compression efficiency of the entire system. If the fact that the compression ratio is improved is previously known from the digital signal sample Y output from the signal source 11, the down-sampling rate in the down-sampling part ~~41~~ 36 can be fixed. However, if only some blocks of digital signal samples X in one sequence require down-sampling, or if it is preferred that the down-sampling rate is changed for each block, as shown by a dashed-line in FIG. 12, a sample sequence dividing part 31 is provided to divide the sequence of digital signal samples X in the floating-point format into blocks of a predetermined number of samples. For each of the divisional blocks, an evaluation part 38 evaluates the case where down-sampling is performed and the case where down-sampling is not performed in terms of compression efficiency of the divisional block, that is, in terms of sum of the number of bits of the code Ca output from the compressing part 13 and the number of bits of the code Cb output from the compressing part 17 to determine which of the two cases provides a smaller sum, determines whether to perform down-sampling in the down-sampling part 36 or whether to increase or decrease the sampling frequency in order to provide a higher compression efficiency, that is, to make the sum of the number of bits of the code Ca and the number of bits of the code Cb, and outputs the codes Ca and Cb that correspond to the better or best

case. Besides, an auxiliary coding part 39 outputs, as an auxiliary code Cd, a code that indicates whether the code Ca to be output has been down-sampled or not or auxiliary information that indicates the down-sampling rate.